Recursive Tree Structure (RTS) forwarding

PIM-WG, IETF118 Prague, v1.0

draft-eckert-pim-rts-forwarding-00

Toerless Eckert, Futurewei USA (tte@cs.fau.de) (Editor)
Michael Menth, (menth@uni-tuebingen.de),
Steffen Lindner, (steffen.lindner@uni-tuebingen.de)

Why are we here ? – FOR YOUR OWN READING

2002?…2014 SPs want MPLS multicast… Industry implements, IETF standardized

After actual deployment experience: tree-state in core too high-opex (large operators)

RFC7988 - Ingress Replication Tunnels in Multicast VPN

Could have stayed with PIM and avoided decade of IETF & industry work on MPLS multicast .. MPLS multicast (RSVP-TE/P2MP, mLDP) also victim of unicast MPLS evolve to SR-{MPLS,v6}.

2013..now: BIER – Bit Index Explicit Replication

"Source-routed-replication" instead of tree state.

Common source-routing header independent of unicast (MPLS/IP)

"global" (domain-wide) bitstrings to encode addresses

Control plane from SR - "do everything needed with IGP extensions"

RFCs (not architecture) focus still on (big) service provider networks(PE to PE).

2015…now: BIER-TE tree engineering with BIER architecture

Utilize same global bitstrings, (almost same) forwarding-plane mechanisms as BIER

2020…now: experimental work looking beyond global bitstrings – for HIGH SPEED HARDWARE

What is the best source-routing mechanism with todays/future hardware (not 2010 hardware).

E.g.: initial experiment "Recursive Bitstring Tree" (RBS) to PIM/IETF113 and then to BIER-WG.

Challenges of BIER / BIER-TE – *global bitstrings*

Processing of large bitstring (BIER allows up to 4096) expensive/complex

Counter-intuitive .. Potentially HW-over?-expensive:
If router has only 16 neighbors, why to process up to 4096 bits of receivers ?

Optimized P4 implementation with intelligent optimization - 256 bit bitstring [Menth23f]

Smaller bitstrings (e.g.: 256 bits) =>

Requires multiple Bitstring in larger networks = multiple packets

Requires (centralized) **SDN mechanisms for bit assignments**

Makes especially BIER-TE complex/hard-to-scale

TE important for high-value traffic (DetNet - bandwidth/latency managemtent)

Even when multicast trees are small (“sparse”) in large networks.

They will statistically require multiple packet copies

BIER/TE Global bitstring(s) Management

Assume more receivers than bits in bitstring. E.g.: 3x as many.

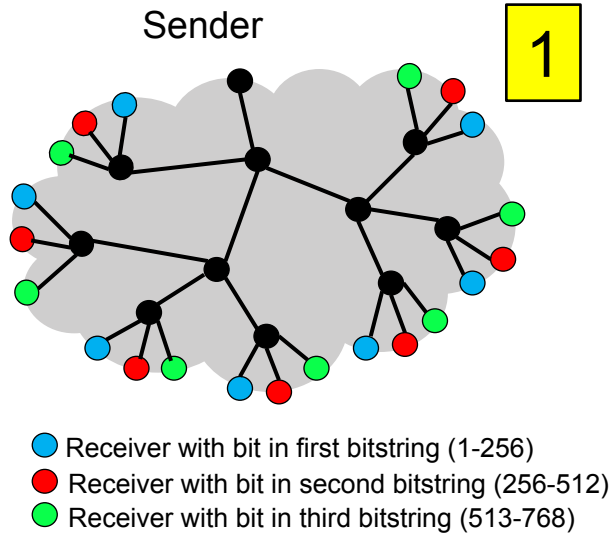
Even just 3 receivers may require 3 packet from the sender

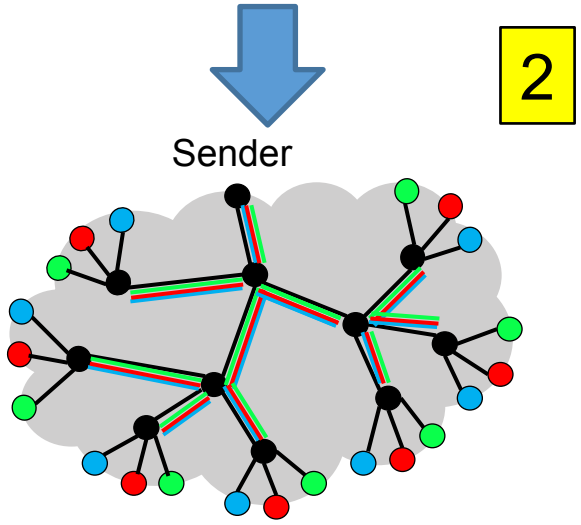
BIER-TE has even more management/scale complexity

- Transit notes needs bits (for steering)
- Need bits in every bitstring (color) passing through them

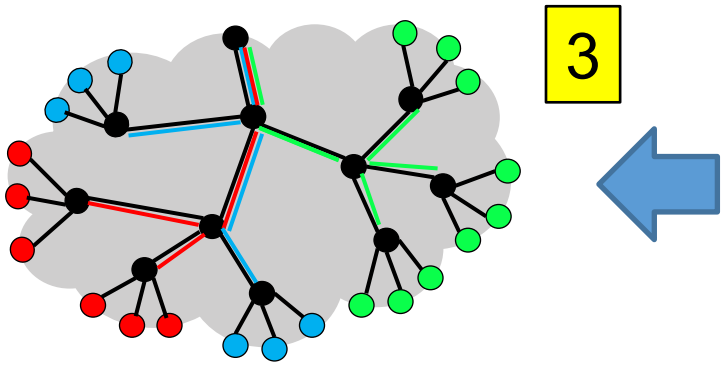
Some scale number presented in MBoned session

Random assignment of bits/bitstring to receivers

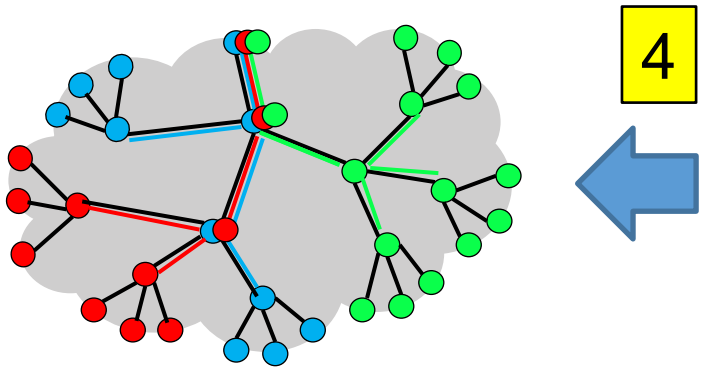




Transit links need to carry up to 3 copies.



Topology aware bit assignment necessary to minimize number of copies



BIER-TE: additional bits for transit notes - multiple!

Solution: Encode the actual (sub) tree in the header

1. Each router only need list of neighbors it needs to replicate to:

   Rx -> {R2, R4} R2, R4 = List of SIDs.

2. Each of those neighbors has their own sub-tree (unless it is a leaf on the tree). We call this sub-tree a Recursive Unit (RU):

   Rx -> { R2 -> RU2 , R4 -> RU4}

   For Rx, the RU for R2 and R4 are just opaque blobs of data.
   There is no parsing of them on Rx.

3. When replicating the packet to a neighbor,
   it only needs to receive the RU designated for it:

   RU2 -> { R5, R6 }

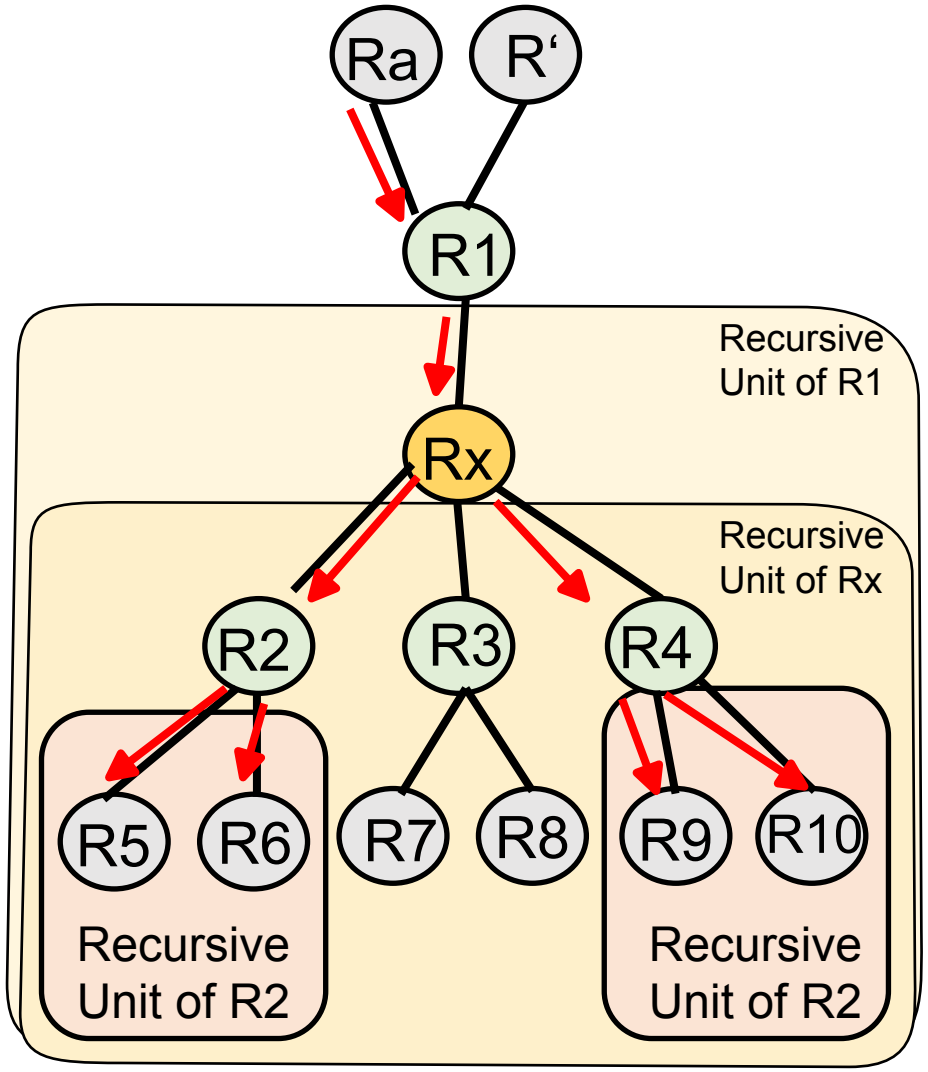
   RU4 -> { R9, R10 }

4. Packet from Ra -> R1 -> Rx:

   Ra -> {R1 -> {Rx -> {R2 -> { R5, R6 } , R4 -> { R9, R10 }}}}

   R1 -> {Rx -> {R2 -> { R5, R6 } , R4 -> { R9, R10 }}}

   Rx -> { R2 -> {R5, R6} , R4 -> {R9, R10}}



Encoding with list of next-thop ‘SIDs’ or local bitstring

Replication with list of SIDs as shown on prior slide:

Rx -> { R2 -> RU2, R4 -> RU4 } RU2, RU4 represent the recursive units for R2, R4

Replication with local bitstring:

Rx -> { [R2, R4] RU2, RU4 } [R2, R4] represent a “**local bitstring**” with bits for all local neighbors of Rx

Which encoding is better / more compact ? *No one-size-fits-all! But many/most trees are sparse*

Assume **local SID** (for all L2 adjacent neighbors) are 8 bit long:

A) Rx is Edge router with 128 neighbors: **local bitstring** needs 128 bit
   local SID list encoding more compact than bitstring if tree has no more than 16 neighbors

B) Rx is Core router with 32 neighbors: **local bitstring** needs 32 bit
   local SID list encoding more compact than bitstring if tree has no more than 4 neighbors

Assume SID list also allows **global SIDs** (16 or 24 bit long).
   Allows to skip over sequences of non-replicating neighbors and produce more compact encodings
   Not applicable where we need “strict trees”, such as when guaranteeing path bandwidth (e.g.: DetNet)

Want an encoding that supports local and global SID lists and local bitstrings !

Forwarding plane implementation considerations

See current state/experiences in BIER-WG presentation

Not yet at implementation stage of supporting local bitstrings, local / global SID lists ***at the same time*** (in P4 on high speed hardware) … But hopeful to get there.

Transferring only RU into packet copies works well.

local / global SID, local bitstrings, RU need to be byte aligned

Across high speed platforms:

Because RU are not examined but only copied

we think this scheme supports largest possible headers

The larger the header, the more receivers a packet will address

This makes large headers always save bits on the wire compared to unicast

No management plane complexity

local SIDs and local bitstring bits assigned autonomously

- Announced into routing (IGP)

Global SIDs are like loop-back interface addresses

- No engineering needed, just uniqueness

Large trees (exceeding maximum header size) can easily be broken into multiple packets / sub-trees – autonomously by library in sender

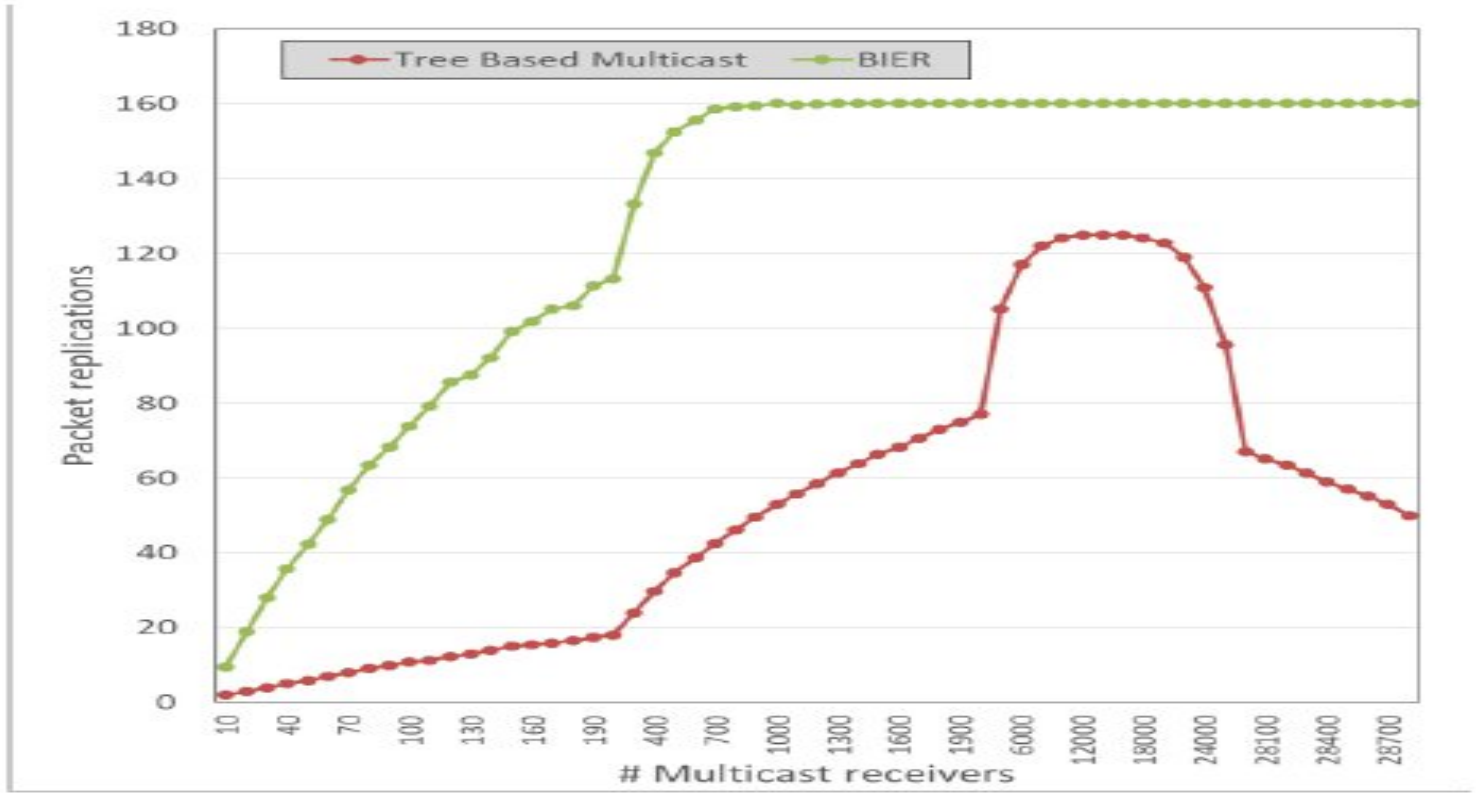
- *Brute force: stop encoding tree when header is full, restart next packet skipping receivers already in prior packets.*

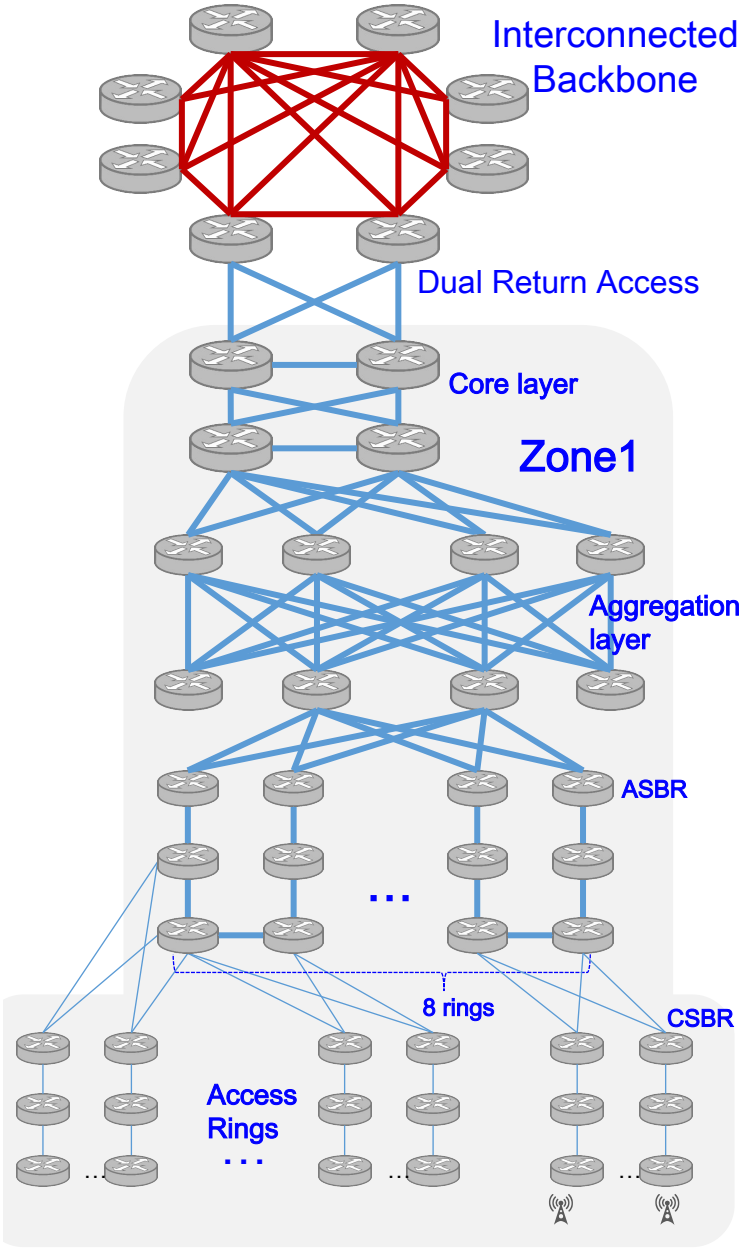
Encoding optimization for large trees

IPTV and other “broadcast” applications may need to
Go to many/most receivers (large tree).

Encoding optimization: “broadcast to all your leaf neighbors”

Simulation comparison BIER, Recursive Bitstrings for large SP
Number of copies even goes down with larger trees!





Where to go from here

Ongoing prototyping work on P4

draft shows hopefully possible encoding ideas

BIER-WG liked initial bitstring encoding (RBS) but said no interest in SID lists.

So not sure which WG to best do this work in.

But not yet asking anyone for adoption anyhow - just looking for interest/collaboration.

Not meant to displace but amend/expand work of BIER/BIER-TE architecture:

Larger networks, less management complexity, better scalable tree engineering, better scalable forwarding plane implementations ???!!!

Broadening interest in stateless multicast source-routing beyond SP would be great!

Presenting to PIM as the larger community